CONTOUR GUIDANCE AND CONTROL SYSTEM ALGORITHM DESIGN AND DEVELOPMENT

Gabe ROGERS
Jason BUNN
Wayne DELLINGER

The Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, Maryland 20723
Gabe.Rogers@jhuapl.edu

ABSTRACT – *The Comet Nucleus Tour spacecraft is the Johns Hopkins University Applied Physics Laboratory's second NASA Discovery series spacecraft. Its goal is to fly by and collect science data on two comet nuclei. This paper presents the design and development of the 3-axis guidance, control, and estimation algorithms, their interaction with the rest of the spacecraft, and the performance of this design. Implementation using The MathWorks' MATLAB/Simulink® and Real-Time Workshop® programs is discussed. The result of this effort is a robust system that meets or exceeds all mission requirements with blocks of code that can be reused for future missions.*

KEYWORDS: CONTOUR, attitude guidance, attitude control, attitude estimation, algorithm design, Simulink, Real-Time Workshop, Kalman filter, thruster control, spacecraft

INTRODUCTION

On 12 February 2001, the NEAR Shoemaker spacecraft ended its historic mission to the asteroid Eros by becoming the first spacecraft to land on an asteroid. NEAR was the first of the Discovery series of NASA missions to be built by the Johns Hopkins University Applied Physics Laboratory (APL). In July 2002, the Comet Nucleus Tour (CONTOUR) mission will build on APL's NEAR success. The CONTOUR spacecraft will embark on a multi-year primary mission to fly by at least two comets. An encounter with comet Encke is scheduled for November 2003, and an encounter with Schwassmann-Wachmann 3 is scheduled for June 2006. A possible extended mission could include flybys of two more comet nuclei. The three primary science objectives are to assess the diversity of comets, to study the processes of comet nuclei and to assess the differences between Kuiper Belt and Oort Cloud comets [1].

To accomplish this mission CONTOUR requires a robust yet simple guidance and control (G&C) system that can perform 3-axis attitude estimation and control. Science and mission objectives require the spacecraft to maintain 3-axis attitude knowledge to 100 μrad, control attitude errors to below 1745μrad, and control rate errors to better than 200 μrad/sec. This paper will describe the design and development of the onboard G&C software algorithms for the CONTOUR spacecraft, and how the software interacts with the attitude sensors and actuators. Also described are the health checking, failure recovery, and safing conditions used in the G&C software. How the system responds to anomalous conditions while maintaining the safety of the mission in lieu of a fully redundant system is discussed. This paper will also include the simulated performance of the all-thruster spacecraft during a simulated encounter. Finally, a brief discussion on the use of MATLAB Simulink and Real-Time Workshop during algorithm development will be provided.

Spacecraft Configuration

CONTOUR is a simple design, with on one articulated mechanism. With the exception of the CONTOUR Remote Imager and Spectrometer (CRISP) scanning mirror, all instrument and antenna pointing is controlled by moving the spacecraft. Figure 1 depicts the CONTOUR configuration. Nine body-mounted solar panels provide electrical power for the vehicle with the sensor suite positioned throughout the spacecraft body. A Nextel and Kevlar dust shield protects the vehicle from particle impacts during comet encounters. In the figure, two of the science instruments are visible: the Neutral Gas/Ion Mass Spectrometer (NGIMS) and CRISP. In addition to science imaging, the CRISP instrument provides feedback control for the guidance and control tasks and will be discussed in a later section.

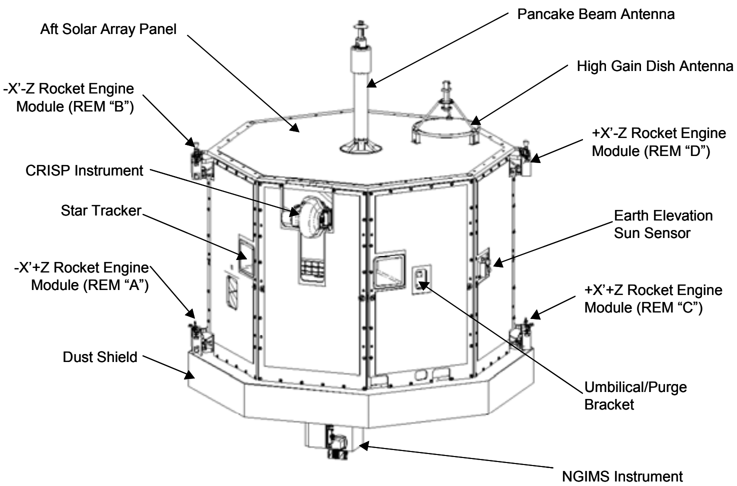


Fig. 1. Spacecraft Configuration

In this paper we will refer to several different coordinate frames, including body frame, propulsion frame, and inertial frame. The body frame consists of the spacecraft Z-axis positive in the direction of the dust shield; and the Y-axis positive in the direction of the CRISP instrument. The propulsion frame is offset from the body frame by -22.5 degrees about the Z-axis such that the X-axis is aligned with the propulsion modules. The inertial frame mentioned throughout this paper is the J2000 TDT equator and equinox frame.

Mission Overview

The CONTOUR mission uses Earth-gravity assist maneuvers to accomplish the multiple encounters. The CONTOUR mission profile is extremely flexible and could be modified to include a first-ever study of a new comet. Contour is scheduled to launch on board a Boeing Delta-7425 during a twenty-five day launch window that opens on 1 July 2002. Following separation, the vehicle enters a series of phasing orbits to adjust its trajectory for a heliocentric insertion burn. The burn will take place in August 2002 to begin the vehicle's orbital tour.

The primary mission consists of two encounters, four Earth flybys, and six periods of hibernation. Figure 2 presents the early portion of the primary mission timeline. In November 2003 CONTOUR will fly within 100 km of comet Encke's nucleus. CONTOUR's second encounter is with comet Schwassmann-Wachmann 3, scheduled for June 2006 and will also have a closest approach range of approximately 100 km. A possible extended mission could include a flyby of comet d'Arrest in August 2008, or the spacecraft could be retargeted to a new, as yet undiscovered comet.

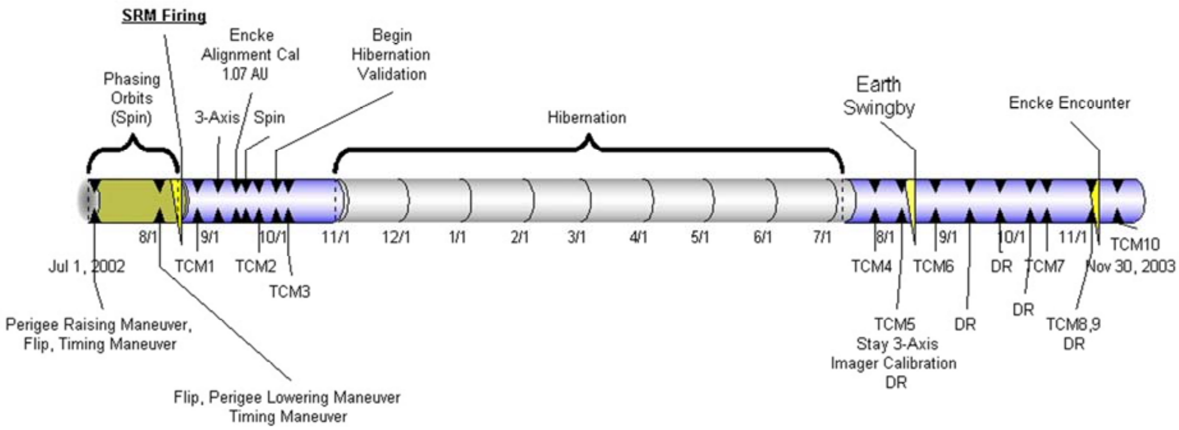


Fig 2. Early primary mission timeline.

SPACECRAFT MODES

Throughout the mission, the spacecraft can be operating in one of several distinct modes. The first is 3-axis/rotiserrie, and is used for instrument calibration, earth flybys, and encounters. 3-axis mode maintains inertial pointing, while rotisserie mode spins about some inertial axis slowly enough to maintain onboard attitude knowledge via star cameras, yet fast enough to provide thermal protection for the spacecraft. Active spin-stabilized mode is used during the phasing orbits, trajectory correction maneuvers (TCM), and cruise phase transitions. Spacecraft control performed open-loop by the ground; the spacecraft is spinning fast enough to provide passive spin axis inertial stabilization.

Finally, during the long periods between Earth flybys and comet encounters the spacecraft will enter hibernation mode, at which point most of the spacecraft systems will be powered down. The spacecraft will spin nominally about its maximum moment-of-inertia at 20 rpm with its spin axis normal to the orbit plane. The G&C system is turned off, so spin momentum stiffness and passive nutation damping from the fuel will maintain spin axis orientation over the long periods of inactivity. Since solar pressure is the only continuous external torque acting upon the spacecraft, the spin axis should precess less than 5 degrees over a 300 day period.

G&C MODES

Spin Mode

Spin mode occurs during the phasing orbits, trajectory correction maneuvers, or as the spacecraft transitions out of hibernation mode. An Earth-Sun sensor is used to measure spin rate and sun angle onboard, but spin axis determination is handled on the ground. The inertial reference unit is turned on to measure rates, but no thruster commands from the G&C are allowed.

Precess Mode

If the sun sensor, while the spacecraft is in active spin mode, cannot detect the sun then the G&C can be commanded from the ground into precess mode. In this mode, the G&C system precesses the spin axis of the spacecraft using gyros to an attitude in which the sun can be detected. Following the completion of this maneuver the ground can reset the spacecraft back into spin mode.

3-axis/Rotiseerie Mode

Earth flybys and comet encounters take place in 3-axis/rotisserie mode. The G&C allows for fine pointing control about some inertial orientation. In addition, this mode has the ability to perform active nutation damping if needed. Transitioning between spin stabilization and active control is handled by this mode. Data downlinks will typically be in rotisserie mode, while science imaging and calibration maneuvers occur using 3-axis control.

G&C SENSORS AND ACTUATORS

During the phasing orbits and cruise phase, while the spacecraft is in spin mode, it is the responsibility of mission analysts to determine the inertial direction of the spin axis of the spacecraft. TCM's and the heliocentric insertion burn require this knowledge to be better than 0.013 rad. During this time the active sensor is the Earth-Sun Sensor (ESS), though onboard gyros can be used to cross-verify the ESS measured spin rate. Spacecraft spin rate and sun angle are computed onboard, but spin axis attitude is determined on the ground using telemetry from the ESS. The ESS was built by Officine Galileo and has flown on more than 70 missions. The sensor has two sun slits and two Earth horizon-detecting telescopes. The telescopes are only used during the phasing orbits. The meridian and skew sun slits have fields of view of ± 80° and ± 60° respectively, thus the ESS has two exclusion zones where it cannot see the sun to perform sun angle and/or sun rate measurements. The spacecraft should be spinning faster than 10 rpm for the ESS to work properly.

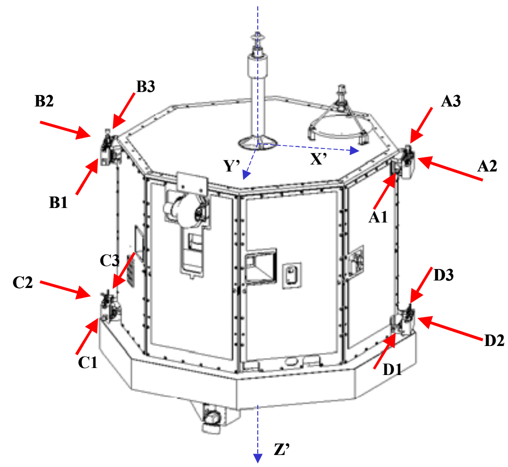
For precess and 3-axis/rotisserie modes all onboard spacecraft attitude guidance, control, and estimation are the responsibility of the G&C system. The 3-axis G&C system consists of two star cameras (ASC) and a single 3-axis inertial reference unit (IRU) for attitude estimation. The ASCs can establish an inertial attitude to an accuracy of 1-2 arcseconds 1-σ in-plane and 5 arcseconds 1-σ about the sensor boresight [2]. The IRU has a bias stability over 1 year of 349 μrad/hr, an angular random walk of 2.9 μrad/sec$^{1/2}$, and a readout noise of 5 μrad. At 100 Hz the IRU can only measure rates up to 31.25 rpm, and for larger rates an overrate flag is set in addition to an overrate bit, which indicates the direction but not the magnitude of spin.

The CRISP instrument and the G&C system pass information back and forth in order for the CRISP to obtain a better track of the comet with its scanning mirror. CRISP reports to the G&C once a second its best estimate of its attitude and the time associated with that message. The G&C in return provides a quaternion correction to that attitude from the estimation task. In addition, the control task reports the commanded guidance quaternion to CRISP. Using the instrument's images of the comet , CRISP will provide a roll correction to the commanded quaternion to better center the comet image in the center of the instruments field of view.

There are a total of 17 thrusters on CONTOUR, including a large solid rocket motor used during the heliocentric insertion burn and 4 monopropellant thrusters only used for trajectory correction maneuvers. General Dynamics supplied the 12 0.8896 N monopropellant thrusters used for attitude control. Their coupled pair configuration provides redundancy about all 3 axes of the spacecraft. At launch the minimum impulse bit of each thruster is 0.05 N-s for a 20 msec pulse width. The propulsion system is simple blowdown, so the minimum impulse bit will decrease with fuel use. To meet science requirements, single thruster control is used to provide torque about each axis during fine pointing control. Figure 3 displays the layout of the attitude control thrusters used in 3-axis mode in the propulsion frame and the default set chosen for single thruster control.

G&C FLIGHT SOFTWARE ALGORITHM DESIGN

The attitude estimation and control activities are two separate tasks. The attitude estimator (AE) runs at 1 Hz and, by processing star camera and IRU data though a Kalman filter, computes vehicle attitude and rate, gyro biases, and star camera misalignments. The attitude controller (AC) executes at 25 Hz and computes commanded and measured attitude quaternions to generate thruster commands needed for controlling the spacecraft's orientation.



TORQUES	THRUSTERS
SPIN UP	C3,D1 and/or B3,A1
SPIN DOWN	C1,D3 and/or B1,A3
- ABOUT X'	C3,B1 and/or D3,A1
+ ABOUT X'	C1,B3 and/or D1,A3
- ABOUT Y'	B2,D2
+ ABOUT Y'	C2,A2

SINGLE THRUSTER CONTROL	
+X'	B3
-X'	D3
+Y'	A2
-Y'	B2
+Z	C1
-Z	C3

Fig. 3. Spacecraft thruster configuration in propulsion frame

Attitude Estimator

During 3-axis mode, the Attitude Estimator receives data from the IRU, the ASCs, the CRISP instrument and the attitude controller. The sensor data consist of time-tagged three-axis angle data (sampled at 100 Hz to form a 300 element time-tagged buffer) from the IRU and two time-tagged quaternions sampled at approximately 2 Hz from each star camera. The CRISP and the controller also send their time-tagged attitude estimates. Internal to the estimator is a 100 Hz loop that processes these data using a Kalman Filter to compute estimates of vehicle attitude and rate, gyro biases and star camera misalignments. During the data processing, the AE computes the error between its attitude estimate and the time-tagged quaternions received from the CRISP and the AC. This quaternion error is sent back to the AC or CRISP for incorporation into their individual attitude propagators.

IRU Interrogation and Attitude Propagation

The first actions of the estimator during each internal 100 Hz loop is to interrogate the IRU data and compute the average rate over the 10 ms since the last sample, and to save the accumulated angle and accumulated time since the last time the Kalman Filter Covariance was updated. The IRU times serve as a marker to judge when the propagated attitude is closest in time to a star camera measurement, the CRISP attitude or the AC attitude. When the time of the propagated attitude matches the measurement or external estimate, a flag is set to trigger either the incorporation of the measurement into the Kalman Filter or the computation of the error between the external reference (CRISP or AC) and the estimator's quaternion.

Kalman Filter Processing

Table 1 lists the standard Kalman Filter Equations used in the CONTOUR estimator. The majority of the Kalman Filter processing simply implements these equations and will not be discussed in detail since rigorous treatments abound in the literature[1]. There are, however, three significant features of the filter implementation to address: the state transition matrix Φ, the use of the form of the measurement matrix H to reduce numerical complexity, and the modification of the H matrix to estimate sensor misalignments.

[1] For one of any number of examples, see reference [3].

Table 1. Attitude Estimator Kalman Filter Governing Equations

$z_k = H_k x_k + v_k, \quad v_k \equiv N(0, R_k)$	Measurement
$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1}$	Covariance Propagation
$P_k(+) = [I - K_k H_k] P_k(-) [I - K_k H_k]^T + K_k R_k K_k^T$	Covariance Update
$K_k = P_k(-) H_k^T \left[H_k P_k H_k^T + R_k \right]^{-1}$	Kalman Gain
$\hat{x}_k(+) = \hat{x}_k(-) + K_k [z_k - H_k \hat{x}_k(-)]$	State Update

State Transition Matrix

The CONTOUR filter takes its state transition matrix from reference [4]:

$$\Phi_{k-1} = \begin{bmatrix} \Theta_{k-1} & \Psi_{k-1} \\ 0_{3x3} & I_{3x3} \end{bmatrix} \tag{1}$$

with

$$\Theta_{k-1} = A(t_k) A^T(t_{k-1}) \tag{2}$$

$$\Psi_{k-1} = \int_{t_{k-1}}^{t_k} A(t_k) A^T(t') dt' \tag{3}$$

where A(t) is the attitude matrix used to transform vectors from inertial to body frame. Assuming a constant rate over the interval [t_{k-1}, t_k], Θ and Ψ become:

$$\Theta_{k-1} = I_{3x3} + \sin(\phi_{k-1})[[\hat{n}_k]] + (1 - \cos(\phi_{k-1}))[[\hat{n}_{k-1}]]^2 \tag{4}$$

$$\Psi_{k-1} = a_{k-1} I_{3x3} + b_{k-1}[[\hat{n}_k]] + c_{k-1}[[\hat{n}_{k-1}]]^2 \tag{5}$$

$$a_{k-1} = t_k - t_{k-1} \tag{6}$$

$$b_{k-1} = \frac{1 - \cos(\phi_{k-1})}{\phi_{k-1}} (t_k - t_{k-1}) \tag{7}$$

$$c_{k-1} = \left(1 - \frac{\sin(\phi_{k-1})}{\phi_{k-1}}\right)(t_k - t_{k-1}) \tag{8}$$

Applying first order and small angle approximations, Θ becomes,

$$\Theta_{k-1} = I_{3x3} + \phi_{k-1}[[\hat{n}_{k-1}]] \tag{9}$$

with the second term of Θ as the skew symmetric form of the accumulated angle vector from the IRU:

$$\phi_{k-1}[[\hat{n}_{k-1}]] = \begin{bmatrix} 0 & \phi_3 & -\phi_2 \\ -\phi_3 & 0 & \phi_1 \\ \phi_2 & -\phi_1 & 0 \end{bmatrix} \tag{10}$$

Eliminating higher order terms and using the small angle approximation, Ψ becomes,

$$\Psi_{k-1} = \Delta t * I + \tfrac{1}{2}\Delta t * \phi_{k-1}[[\hat{n}_{k-1}]] \tag{11}$$

We can now form Φ at each time step. However, this assumes the state is only the three angular estimates and the three rate estimates. Contour also estimates the gyro biases and the misalignment between the two star tracker heads. There is also the issue of the IRU measurements and their effect on the Φ matrix to consider.

The Kalman filter tracks gyro biases and then updates the estimate of the vehicle rate with these biases to better track the true rate. Assuming the gyros are functional, the Φ matrix is modified by an alignment matrix that transforms gyro bias state into the vehicle frame. This augmented Φ has the form:

$$\Phi = \begin{bmatrix} \Theta & \Psi & ALNG \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \tag{12}$$

If the gyros are not functioning, this ALGN matrix is set to zero.

An identity matrix is also added to the Φ matrix to include the star tracker misalignments in the state vector:

$$\Phi = \begin{bmatrix} \Theta & \Psi & ALGN & I \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{14}$$

The Φ matrix we have assembled constricts the format of the state vector. We now must have three attitude measurements first, followed by three rate measurements. Some number of gyro misalignments are next, followed by three elements of one sensor-to-sensor misalignment.

The gyros are measuring rate, while the Kalman Filter is estimating rate. Ideally, we would like to use the gyro reading as the "true" rate. This corresponds to a perfect gyro measurement. If we were to simply proceed with perfect gyro measurements, there would be no growth in the attitude variance due to the rate state. This is not, however, what is desired if the gyros are turned off or failed. Therefore, the following procedure, which was developed for the NEAR mission [5], is employed on CONTOUR:

- Form the Φ matrix as in (14) and compute $\Phi P \Phi^T + Q$ following the standard Kalman filter formulation

- If the gyros are on, decorrelate the gyro states from the rest of the states by setting sections of the covariance matrix to zero. For example, if our state consists of attitude, rate, bias, and alignments, in that order, the following procedure accomplishes decorrelation:

$$P_{k+1} = \Phi P_k \Phi^T + Q = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{12} & P_{22} & P_{23} & P_{24} \\ P_{13} & P_{23} & P_{33} & P_{34} \\ P_{14} & P_{24} & P_{34} & P_{44} \end{bmatrix} \Rightarrow \begin{bmatrix} P_{11} & 0 & P_{13} & P_{14} \\ 0 & P_{22} & 0 & 0 \\ P_{13} & 0 & P_{33} & P_{34} \\ P_{14} & 0 & P_{34} & P_{44} \end{bmatrix} \qquad (15)$$

- Performing this decorrelation after the propagation allows us to place some small value for gyro noise in the covariance matrix (element P_{22}, above). This noise will provide the desired uncertainty in the IRU rate, but then the decorrelation will prevent the Kalman filter from actually updating the rate state estimate, which we take as simply the gyro measurement. This procedure allows us to incorporate the high frequency gyro measurement into the processing without performing a formal filter update each 100 Hz cycle.

- If the gyros are not on, the decorrelation does not happen, allowing the natural correlation between position and rate to take place and the rate state, updated by the star tracker measurement, is used as the rate estimate.

Computing Sensor Misalignments

When a star tracker measurement is nominally incorporated into the Kalman Filter, the quaternion is compared to the AE estimate, an error in radians is computed, and then the state update equation is applied. Thus the actual state is the expected angular error between the star tracker quaternion and the estimated quaternion. To estimate sensor misalignments, the H matrix is modified so that the measurement from ASC2 is defined to be the sum of the angular error and the misalignment of star camera 2 from its expected position. Thus the quaternion product of the attitude and misalignment estimate is compared to the measurement. This error signal is used to update the state vector.

Numerical Complexity Reduction

The CONTOUR estimator also takes advantage of the form of the H measurement matrices to reduce the complexity of the Kalman Filter equations. As stated earlier, the measurements for CONTOUR consist of inertial rate information from the IRU and an attitude reference from the ASCs. Since the IRU rate data are not measurements to be incorporated into the Kalman Filter, the only measurement we need be concerned with here is a 3-element attitude measurement for tracker 1 and a six-element attitude and misalignment measurement for tracker 2. Re-arranging the state vector to have the attitude states and the misalignment states as the first six elements, the H matrix would only have information in the first three (or six) out of s columns, where s is the number of states. This re-arrangement of the state vector also rearranges Φ in (14).

Exploiting this fact in the Kalman Gain and Covariance Update Computations resulted in a substantial operations savings. The Kalman Gain complexity was reduced from quadratic to linear in the size of the state vector, and the Update Covariance complexity was quadratic instead of cubic.

Fault Detection and Mitigation

The final portion of the Estimator design was the implementation of fault detection and mitigation algorithms. The vast majority of these algorithms concern checking the incoming sensor data for goodness and proper format. The remaining algorithms verify proper functioning of the Kalman Filter. Table 2 lists the different fault detection algorithms.

Table 2. Attitude Estimator Fault Detection Algorithms

Sensor	Fault	Response
IRU	Validity flag not set	Reject IRU data until two consecutive good readings received
	Time difference between consecutive 100 Hz samples out of bounds	Reject IRU data
	Time indicates stale data	Reject IRU data
	Counts indicate angle too large	Reject IRU data
	Counts indicate stale data	Begin counting instances of stale data, reject if threshold exceeded
ASC	Quaternion Norm Invalid	Reject ASC measurement; do not incorporate into attitude estimate
	Validity or Quality Flags invalid	Reject ASC measurement
	ASC measurement stale	Reject ASC measurement
CRISP	Invalid Norm	Do not perform correction; output [0 0 0 1] quaternion correction
Kalman Filter	Failed matrix inverse in Kalman Gain Computation	Do not incorporate measurement
	χ^2 test on residuals failed. (test on the weighted norm of the filter residuals, compared to tabulated confidence values)	Do not incorporate measurement

If the IRU or ASC fault detection algorithms are triggered, counters track the instances of consecutive failure and report that information to the ground. In addition, the Estimator reports both an Attitude Quality and an Attitude Knowledge bit that summarizes the health of the Attitude Estimator. The quality can range from 0 (no sensor data and no confidence in attitude estimate) to 3 (majority of rate processing from IRU, at least one star camera processed each second). The knowledge flag is a one bit indicator that is set if the quality is 1 or above, indicating some confidence in the AE attitude estimate.

Attitude Controller

The responsibility of the AC is to provide thruster commands to the spacecraft to control attitude and rate, as well as to provide an estimate of rate to compare with the ESS. Thruster commands are output 25 times a second, or once each AC control cycle (ACCC). The AC task uses IRU data in conjunction with attitude quaternion corrections from the AE to generate its estimate of the current attitude. In addition, the AC contains the guidance algorithms needed to generate a commanded attitude quaternion and rate. The guidance system algorithms are based upon NEAR heritage [6].

IRU Message Processing

The IRU measures the rotational rate of the spacecraft at 100 Hz in the form of delta angles. The 100 Hz IRU message is buffered and sent to the AC at 25 Hz. Each ACCC the 4 buffered angle measurements and the last measurement from the previous ACCC are used to calculate 4 rates. Checks on the incoming measurements include stuck axes (non-changing data), delta angle too large, and bad delta time. If any of these checks indicate a bad data point then the rates calculated using that point are discarded. The remaining rates are averaged and the result is an unfiltered AC measured rate for the current ACCC. If an overrate bit is set for any one of the 5 IRU measurements, or if none of the computed rates were valid, then the IRU data is flagged bad and will not be used for control.

Attitude and Rate Computation

Unfiltered AC measured rates can be used for coarse rate-only control, but are too noisy to be used for science pointing. The gyros, whose output is in accumulated angle, have a read out noise of 5 μrad. Thus, if only one rate per cycle was valid for the 100 Hz signal the unfiltered rates could have noise of up to 1000 μrad/sec 3-σ. The normal noise using all 5 measurements however is much smaller than this value, approximately 250 urad/sec 3-σ. Because science requirements constrain rate control to 200 μrad/sec the rates must be filtered. The unfiltered rates are passed through a second order Butterworth low-pass filter. Filtered IRU noise can produce a rate signal to 35 μrad/sec 3-σ with a lag of approximately 90 msec. This is sufficient for science pointing, though steps to account for the lag must be taken. Figure 4 demonstrates the effect of filtering on measured rates.

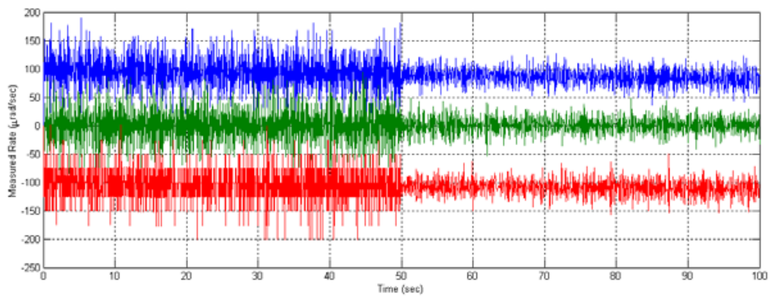


Fig 4. Filtered and Unfiltered AC Measured Rates.

Estimated attitude is computed by integrating the unfiltered AC measured rates, so as to not introduce the filter lag into the computation. Once a second this attitude is passed to the AE task, which compares the AC attitude to the AE attitude. A quaternion correction is then passed back into the AC from the AE, which is immediately applied to correct the AC's estimate.

A final function performed by the rate computation algorithms is to compare the filtered measured rates with the ESS estimate of rate. If the difference between these two values is small, then an ESS/IRU discrepancy flag is set to ok. If not, than an error is set in housekeeping telemetry.

Guidance Algorithms

The guidance algorithms are used to orient the spacecraft to a particular commanded quaternion, at a commanded rate. The resulting quaternion is in the propulsion frame. The commanded quaternion for non-spinning 3-axis control is generated using four vectors: aimpoint, auxiliary roll vector, virtual boresight, and roll reference vector. Aimpoint and auxiliary roll vectors are specified in an inertial direction, while virtual boresight and roll reference vectors are specified in the body frame. To generate the commanded quaternion the spacecraft aligns the virtual boresight to the aimpoint. To determine the amount of roll about the virtual boresight requires both auxiliary roll and roll reference vectors. The commanded quaternion places the roll reference vector into the plane formed by the aimpoint and auxiliary roll vectors. A final roll correction quaternion provided by CRISP is multiplied with the commanded quaternion to fine-tune the spacecraft's attitude during encounters.

The aimpoint and auxiliary roll vectors can be specified as either a ground uploaded inertial vector, a vector to a central body, or the reference velocity vector. The central body vectors include vectors from the spacecraft to Earth, Sun, or comet. These vectors are generated by approximating DE-406 positions into Chebyshev ephemeris coefficient polynomial sets. The velocity vector is the direction of spacecraft motion in the inertial reference frame. The ground-uploaded aimpoint can be specified in two ways: the first is as a J2000 unit vector centered on CONTOUR; the second is as a vector from the comet to an inertial reference point in the Comet Centered Inertial frame. An unloadable parameter lets the AC differentiate between the two. If the aimpoint is specified as a vector from the comet to an inertial point then the AC converts this vector to a J2000 unit vector centered on CONTOUR pointing to the reference point.

The values for the four guidance vectors are dependent on a ground commanded guidance scenario. There are a total of 16 different guidance scenarios the spacecraft can be commanded into by mission operations. Scenario 0 is a fixed quaternion, which bypasses all of the guidance algorithms to force the commanded quaternion to a specific value. Scenarios 1-4 are default scenarios used to protect the spacecraft during encounters or anomalies. Scenarios 5-15 are specified by the ground, and can be modified through 7-element uploadable parameter blocks to create any commanded attitude desired by science or mission operations.

It should be noted that the process used to generate the 'before CRISP' unmodified commanded quaternion from the four input vectors and the guidance scenario actually takes place in three steps. Step one generates a quaternion transformation using only the inertial vectors (aimpoint and auxiliary roll). We term this the inertial to propulsion prime quaternion. The transformation aligns the Z-axis of the spacecraft propulsion frame with the inertial aimpoint, and places the Y-axis of the propulsion frame towards the auxiliary vector in the plane formed by the two before mentioned inertial vectors. Step two involves determining the default virtual boresight and roll reference vectors for the desired guidance scenario. There are 15 different virtual boresights and roll reference vectors for guidance scenarios 1 through 15. The desired scenario determines which boresight and roll reference vector is used. Step three involves generating a propulsion prime to propulsion quaternion using these virtual boresight and body fixed roll vectors. A quaternion multiply between the step one and step three quaternions completes the commanded quaternion. It is this final quaternion, multiplied by the CRISP correction quaternion, which is used for control.

To protect the spacecraft, two closest approach timers have been added. These timers are a countdown until the time of closest approach as estimated by the C&DH and CRISP. The C&DH's timer is computed using the ground's best estimate of the spacecraft position with respect to the comet. CRISP's timer is computed using optical navigation images to compute a pseudo real-time estimate of spacecraft position. When either timer decrements below a set threshold, then a bypass is tripped to force the spacecraft into a safe attitude for the flyby. The bypass consists of using the propulsion prime quaternion multiplied with a fixed, default quaternion to generate our 'before CRISP' commanded quaternion. The default quaternion assures that the spacecraft will have the Z-axis aligned with the velocity vector, the safest condition for the spacecraft, and leave the spacecraft Y-axis pointing towards the comet, so that CRISP can perform imaging. During non-encounter maneuvers both the CRISP and C&DH closest approach timers can be disabled.

There are a couple operational constraints to this methodology that should be kept in mind. Once we enter rotisserie mode the spin vector remains inertially fixed, which prevents us from tracking a moving body such as the Earth. Also, if operators want to change the inertial spin vector, the commanded rates must first be zeroed.

Precess Control

Precess control is only used when the spacecraft is spinning faster than 10 rpm and cannot determine the spin rate and/or sun angle from the ESS. When the AC is commanded into precess mode it uses the measured IRU data to determine the spin rate. If the rate is too low or too high then the spacecraft spins up or down to a parameterized range. Once in range, the IRU rates are integrated to obtain a delta angle, initialized when the precess command is initiated. Two sequences of thrusters then fire for a specified time 180 degrees apart in the spin cycle to precess the spin axis. We do not care which direction inertially we precess, only that the spin axis is moving in a constant inertial direction. Eventually, the spin axis will move enough for the ESS to compute spin rates and sun angles. When the sun angle is below a certain range, a precess complete flag is sent and any further thruster signals are nulled. The precess flag is a semi-sticky flag (held constant for a specified period of time), so if nutation forces the spin axis outside the parameterized range the thrusters will not start to fire again.

3-axis/Rotisserrie Control

The purpose of the 3-axis control algorithms is to align the measured AC attitude and rate to the AC commanded attitude and rate. For nominal control this is accomplished using a phase plane controller. In addition the AC has the ability to provide active nutation damping. For this a proportional-derivative controller is used. The result of each controller is a command torque, which is passed to a thrust command generator. The thrust command generator creates a sequence of on/off commands for each attitude thruster every ACCC based on the commanded torque.

The phase-plane controller contains several different sets of switching lines, depending on the health and state of the system. The intent is to allow the spacecraft to meet the science and mission requirements while minimizing fuel usage. In addition to controlling attitude and rate errors to tolerances mentioned earlier, the spacecraft also must meet safety requirements of recovering from a 5 mg particle impact at 28 km/sec on the dust shield in less than 4 seconds. Four sets of switch lines are contained in the phase plane controller: fine pointing, slewing, rate-only, and 1 Hz. Figure 5 presents the slewing and fine pointing switching lines. In this figure the square is defined by the slew attitude and rate thresholds.

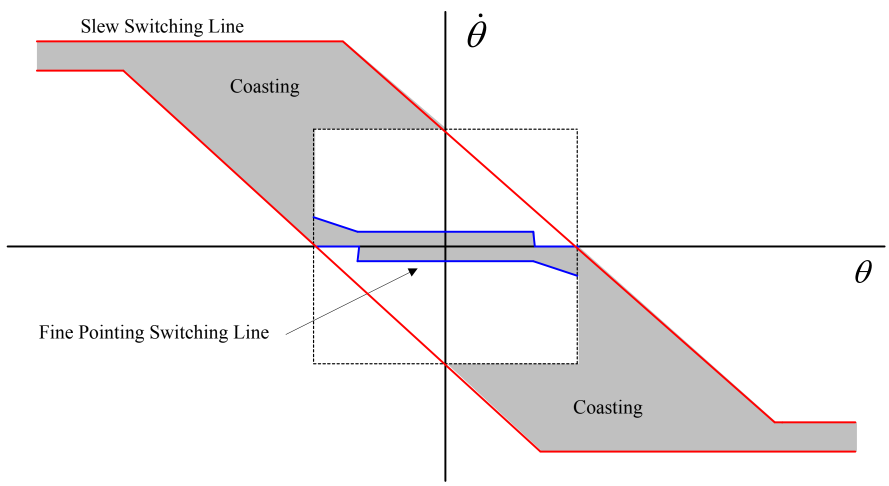


Fig. 5. Normal 3-axis phase plane slew and fine pointing switching lines.

Slew switching lines are used when the measured attitude or rate errors exceed a parameter threshold. If they do, then a slew flag is set to true and the spacecraft attempts to push these errors below the slew threshold. These lines also contain a maximum slew rate to protect against a loss of communications during a slew. There is a rate deadband such that when attitude error is large the spacecraft will coast towards the commanded attitude, minimizing fuel use. Fine pointing switching lines are used when the slew flag is set to false. These lines force the spacecraft attitude and rate errors to below the science requirements. Rate only switching lines only look to see if the AC rate error is above or below a specified threshold, and try push the errors to within this value. 1 Hz switching lines are used when the IRU is flagged bad but the AE is still producing a valid attitude and rate estimate. These switching lines are identical to the fine pointing lines except they prevent the spacecraft from slewing faster than a set value to prevent the star cameras from loosing track.

The thruster command generator takes the control torques and forms a sequence of thruster commands. The values in the command depend on two factors; the slew-in-progress flag and if the AC is in 1 Hz mode. If either of these is true then the AC will generate a command which fires coupled thrusters for the entire length of the ACCC. If neither of these are true then the AC will generate a command to fire a single thruster for one half ACCC. These commands will be overridden by a third sequence if an IRU overrate bit is set true.

The results of the thruster command generator are then passed though a duty-cycle generator. This code forces the AC to wait a set number of ACCC after a thruster firing before allowing another firing. This is to account for the time lags introduced by the thruster relays and low-pass IRU filters. The duty-cycle generator is set to 100% when the slew flag is set true. For normal fine pointing operations this duty cycle is set to 20%. For 1 Hz control the duty cycle is set to 1%. The delays between firings help to prevent multiple firings for the same event.

To moderate thermal conditions on the spacecraft, CONTOUR can be placed into a slowly spinning, 3-axis mode termed Rotisserie Mode. The spin rate is slow enough that the star cameras can still maintain a track of stars, but fast enough to insure the proper thermal conditions throughout the spacecraft. For this reason, in rotisserie mode we control using both the non-zero rates as well as a commanded quaternion. The rotisserie commanded quaternion is calculated each ACCC and is generated by taking the current commanded quaternion and propagating it forward one ACCC using the commanded rates. This new quaternion will then be used during the next control cycle to generate the following quaternion. This process repeats until the commanded rates are set to zero.

Thruster Safety Checks

There are 3 internal checks the AC performs to insure thruster commands do not interfere with science imaging or possibly damage the propulsion system. To insure the integrity of CRISP and CFI images, the AC can be commanded to inhibit all thruster firings for a specified period of time each second. To protect against using a bad thruster, a thruster mask is commanded to inhibit individual thrusters. Finally, to protect against overheating thruster valves, a timer prevents any thruster for firing more than ten minutes at a time.

Attitude Controller Heath Checking

There are additional health checks the AC produces to inform mission operations of the status of the AC system. The AC controller health is checked to see how well the AC is able to maintain control of the spacecraft based on sensor health and data availability. This 2-bit flag informs an operator if the controller is healthy and controlling both attitude and rates, healthy but only controlling rates, controlling the spacecraft in 1 Hz mode, or not controlling at all due to a failure of the IRU and AE quality. Ephemeris health checks determine if the current computed time (in TDT) is within the range of the Chebyshev polynomials. ESS health flags inform the operators if the spin rate and sun angle calculations are valid or invalid. The CRISP flag determines if the CRISP message is invalid or disabled. Finally, a thruster health bit attempts to inform operators of a failed thruster. Because there are no accelerometers on the spacecraft, this flag is dependent on whether the spacecraft maintains operational specifications over a set period of time. If not, then the flag is set to invalid. This flag is disabled if the spacecraft is not in 3-axis/rotiserrie mode or is spinning faster than 5 rpm, as it is when spinning up or down.

G&C SIMULATION PERFORMANCE

In order to demonstrate that the algorithms perform as expected, many scenario simulations have been run using mock-ups of the sensors, C&DH, and space environment dynamics. Presented in this section are just a few representative examples of the performance of the spacecraft during comet encounter conditions.

Figure 6 demonstrates the convergence of the AE residuals following a large slewing maneuver. At the start of this example the spacecraft is slewing such that the star camera image is smeared and they have lost track. As the slew concludes the star cameras regain track at 91 seconds and the residuals quickly converge to a solution. Notice that the lower plot is a continuation of the upper plot, but at a different scale.

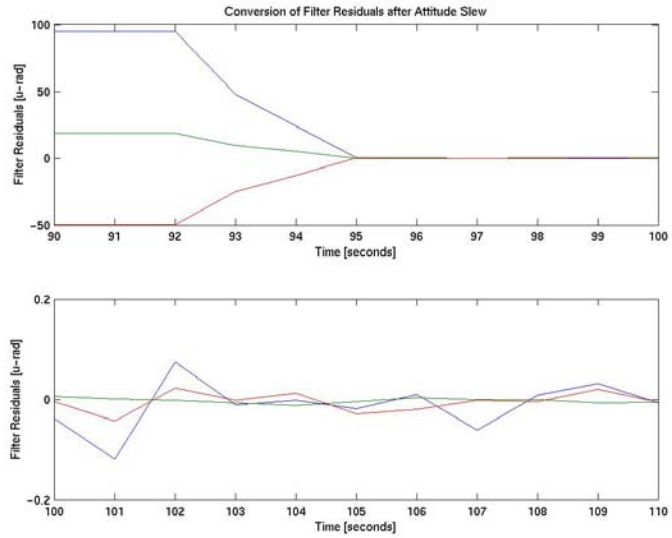


Fig. 6. Conversion of Filter Residuals after slew.

The second simulation is the case where the spacecraft is targeted at a stationary point for an extended period of time, and simply bounces around the attitude and rate deadbands. Artificial IRU biases and star camera misalignments were added to the system to verify the AE solution converged on the correct solution. Figures 7 and 8 demonstrate the performance of the AE and AC tasks. The AE estimates attitude with an error of within 20 μrad 3-σ, which is well within the 100 μrad requirement. The AC is easily able to maintain attitude error to within 1745μrad.

One problem that our simulations uncovered with the AC system is that the thrusters are overpowered for the type of rate control we are trying to maintain. For this reason we had to go to single thruster control when we are in fine-pointing mode. Because of this, thruster torques about the X-axis and Z-axis are coupled. Thus, a firing to correct an X-axis error could introduce a large Z-axis error. For this reason we do not always meet the 200 μrad/sec deadband. However, the system will quickly correct this, so for a large portion of time the spacecraft is meeting rater performance. Figure 9 presents an example of the

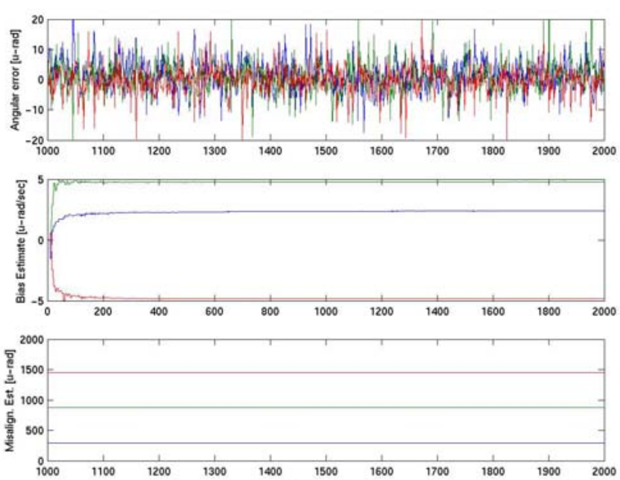


Fig. 7. Attitude estimation calculations during encounter.

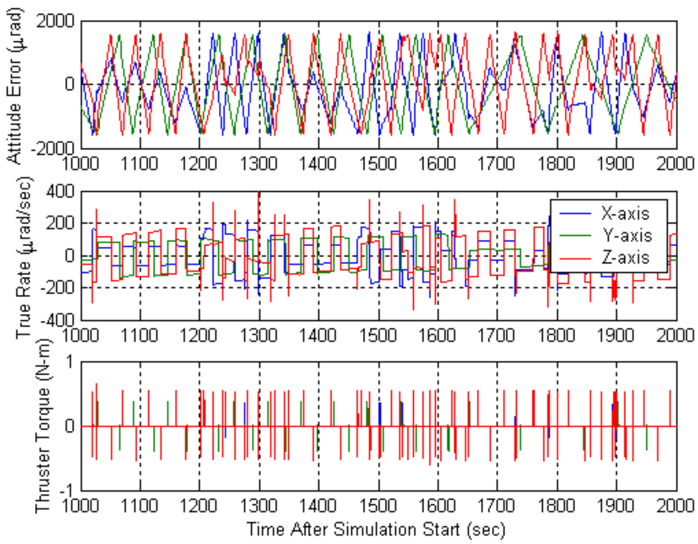


Fig. 8. Attitude errors, true rates, and thruster torques during encounter.

rate performance during inertial targeting. As can be seen, the performance of the AC meets science requirements greater than 96% of the time, and the science team assures us this is sufficient. The performance improves as the feed pressure drops.

MATHWORKS SIMULINK® AND REAL TIME WORKSHOP®

As with the APL developed TIMED spacecraft, the primary design tools for the CONTOUR G&C software were The MathWorks' SIMULINK® and Real-Time Workshop® packages. SIMULINK® utilizes a block diagram construct to build complex systems that can incorporate elementary functions, pre-packaged "black boxes" such as a filter, or custom software modules from other programming languages such as C or FORTRAN. The modules are assembled together in the same way an analyst would draw a system block diagram.

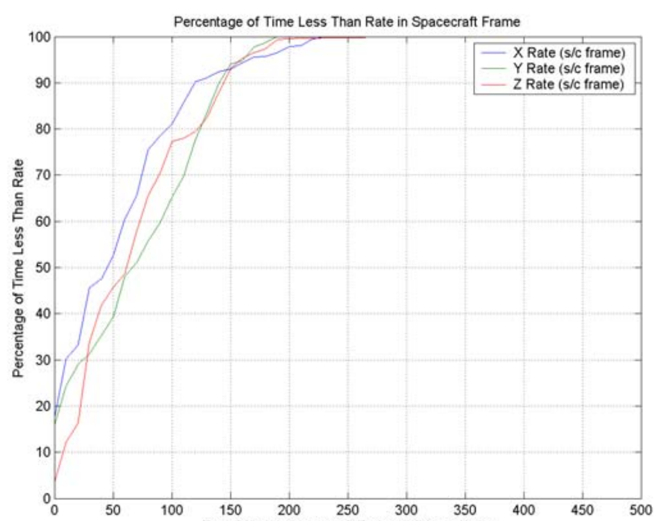


Fig. 9. Percentage of time less than rate in body frame.

The CONTOUR software was developed in a large SIMULINK® block diagram that consisted of not only the fight software, but also the truth dynamics and sensor/actuator models and simpler models of the Command and Data Handling Subsystem and databus transactions. There were several advantages to designing the software in this environment, as well as a few challenges.

The primary advantage of using SIMULINK® was that the interactions between subsystems were greatly simplified. If one software module needed information from another, then the other action necessary was to draw a line from one module to the other and the connection was established. Errors in datatypes, where one module expected an integer and was provided a floating-point variable, for example, were immediately revealed. Testing became a simple exercise of linking test data to the module being tested by simply drawing a line, leading to simple unit testing scenario execution.

Creating subsystems in SIMULINK® was also simple. If a series of modules, or blocks, were best understood as part of a single unit (the gyro processing and IRU health checking modules forming an IRU block, for example), a simple keystroke grouped the series of modules into one block that contained the previous modules at a lower level of detail. It was simple to change the underlying blocks if the need arose. The subsystem construct greatly simplified the SIMULINK® diagrams.

SIMULINK® was not without its challenges, however. The biggest challenge came in how to handle the Attitude Estimation software. When flight software development began, it was assumed that all software would be developed in SIMULINK® blocks. However, when the design of the estimator was considered in further detail, this proved to be exceedingly difficult. As mentioned, the design dictated a need for a 100 Hz loop to run within the 1 Hz Estimator. Each second, the estimator received the buffered data from the previous second and processed these data through a 100 Hz loop, finishing before the next second's worth of data were ready. The difficulty arose because looping in this fashion was not readily available in the version of SIMULINK® with which development began. Subsequent versions of Simulink did support "For" and "While" loops, but those innovations were released too late in the CONTOUR development cycle to be used efficiently. Thus, the vast majority of the Estimator code was hand generated in C and inserted into the overall flight software SIMULINK® diagram in the form of an "S-Function," a SIMULINK® construct that merges SIMULINK® diagrams and custom code. This decision permitted development to continue, but it was not without consequences.

The first consequence was that the inner workings of the estimator were much more difficult to see and thus to debug compared to other portions of flight software. A page of SIMULINK® blocks can be easily exposed to reveal the value of every signal. C code does not have that liberty and more primitive debugging techniques were employed. A second consequence was that any health checking algorithm that had to compare data from within the internal 100 Hz loop had to be inside the custom C code, making the process invisible to the user. Thus most of the IRU health checking, such as examining data for out of bounds conditions or stale data, is buried within the C code. This broke the health checking into two disjointed areas—outside and inside the custom code—and made it difficult at times to isolate where the fault detection algorithms were detecting problems.

When the software was ready to be compiled for execution on the flight hardware, we turned to another The MathWorks' package called the Real-Time Workshop. Real-Time Workshop takes a SIMULINK® model and converts it to the high level language of your choice such as C or FORTRAN. For CONTOUR, the automatically generated code from RTW was then delivered as the flight software. RTW worked extremely well for the program. When the software was ready for delivery, the RTW engineers would copy the Flight code out of the overall SIMULINK® diagram (containing the testbed and other non-G&C flight code) and paste it into a separate SIMULINK® model. Then RTW would convert the new model into flight code. Currently a version of the flight software is able to go from the large SIMULINK® model to 'C' code on the flight computer in under 30 minutes.

CONCLUSION

The CONTOUR G&C team at APL has designed a robust and efficient software package that meets all mission requirements and has passed all phases of testing to date. Building on the experience gained from TIMED and NEAR, the software consists of a 25 Hz Phase Plane attitude controller and a 1 Hz Kalman Filter based attitude estimator. The software was developed using The MathWorks' commercial off-the-shelf tools that provided the needed flexibility for the design effort. Simulations in both the development environment and on the spacecraft hardware provide confidence that the algorithms are properly implemented and will fully satisfy all mission requirements when CONTOUR is launched in July, 2002.

REFERENCES

[1] Dunham, D., Reynolds, E., Chiu, M., and Farquhar, R., CONTOUR – "a COmet Nucleus TOUR", *Proceedings of the 51st International Astronautical Congress*, Rio de Janeiro, Brazil, 2000.

[2] Jøgensen, J.L., "In Orbit Performance of a Fully Autonomous Star Tracker", *Proceedings of the 4th ESA international Conference on Spacecraft Guidance, Navigation and Control*, ESA SP-425, Noordwijk, The Netherlands, 1999.

[3] Gelb, Arthur, editor. Applied Optimal Estimation. The Analytic Sciences Corporation, 1974.

[4] Fisher, H.L., et al, "Attitude Determination for the Star Tracker Mission", *AAS/AIAA Astrodynamics Specialist Conference,* Stowe, Vermont, 1989.

[5] Strikwerda, T.E., et al, "NEAR Guidance and Control System", *Proceedings of the 20th Annual AAS Guidance and Control Conference*, AAS 97-077, Breckenridge, Colorado, 1997.

[6] Heyler, G.A., and Harch, A.P., "Guidance and Control Aspects of the Mathilde Fast Flyby", *Proceedings of the 21st Annual AAS Guidance and Control Conference*, AAS 98-071, Breckenridge, Colorado, 1998.